

[illegible]

SELECTING RENDERING INTENTS

EL 399755643 US

SELECTING RENDERING INTENT

TECHNICAL FIELD

This invention relates to a method for selecting a suitable rendering intent to map the colors of an image created, displayed, or produced in one color gamut for display, production, or reproduction in another color gamut.

BACKGROUND

Color devices, such as color monitors and color printers, are limited in the number of colors they can produce. The range of colors a given color device is capable of producing is known as that device's color gamut.

When a colored electronic document or image having colors contained within one color gamut, is reproduced on or displayed in a device having a different color gamut, color reproduction problems can occur. For example, some of the colors in a document created by selecting colors from the color gamut of a display monitor may not be available in the color gamut of a printer to which the document is sent to be printed. Similarly, a color represented as the color triplet (R_1, G_1, B_1) on the color monitor, may be reproducible on a color printer, but from the color triplet (R_2, G_2, B_2) rather than the triplet (R_1, G_1, B_1) .

Rendering intents are algorithms which have been developed to alleviate such color reproduction problems by suitably mapping colors from the color gamut of one device to the color gamut of another device. Four such rendering intents are commonly available, and are referred to as perceptual, saturation, absolute colorimetric, and relative colorimetric rendering intents. Other rendering intents are possible, however, and any algorithm which defines a transformation or mapping of colors from one color gamut to another color gamut is a rendering intent.

While rendering intents can alleviate color reproduction problems, they do not solve them. Thus a color which is within the color gamut of one device but outside the color gamut of another device cannot be faithfully reproduced on the second device, irrespective of the rendering intent chosen. Moreover, different rendering intents will produce different colors on the second device as they attempt to approximate the out of gamut color.

A user who is sensitive to the color content of a document or image will be sensitive to the various color approximations produced by different rendering intents as they transform

colors from one color gamut to another. Such a user may want to compare the effects different rendering intents will have on the colors in their documents or images prior to selecting one of the intents to transform their documents. Currently, users may only compare the effects different rendering intents have on the colors in documents indirectly. For example, they can select a rendering intent, preview a rendered image, and then either accept the rendering intent or select an alternate rendering intent.

The current process is cumbersome, requires the user to anticipate whether a current rendering intent is better than one which has yet to be viewed, and also requires the user to remember large amounts of information such as subtle differences in color between different rendered images. A method is needed for selecting a desired rendering intent which is less cumbersome, does not require anticipating the quality of images which have yet to be viewed, and does not require the user to remember large amounts of subtle information.

SUMMARY

The invention discloses a rendering intent selection engine for selecting a rendering intent to render the colors of a source document or image having colors contained entirely within a source color gamut so that the document can be displayed, printed, or otherwise output to a destination device having a different destination color gamut.

The rendering intent selection engine receives a color document or image and a set of rendering intents. It renders the image according to the received rendering intents, and simultaneously previews the rendered images. It receives a selection from among the simultaneously previewed rendered images, and selects a rendering intent based upon the received rendered image selection.

The rendering intent selection engine can receive an entire document or image, or only a portion of an entire document or image. It can receive user input specifying the portion of a document or image to be rendered. Similarly, the rendering intent selection engine can receive all known rendering intents, or a subset of all known rendering intents. It can receive user input specifying the rendering intents to be used to render the received image.

The rendering intent selection engine can simultaneously preview the rendered images by simultaneously displaying them on a monitor. Or, it can simultaneously preview

the rendered images by printing them on a single sheet of paper. Or it can display the color values associated with a rendered image on a monitor, or print them on a single sheet of paper. The rendering intent selection engine can preview differences in rendered images rather than the images themselves. It can preview the differences by simultaneously displaying them on a monitor, or printing them on a single sheet of paper, or by simultaneously displaying the difference values on a monitor or printing them on a single sheet of paper.

The rendering intent selection engine creates the difference images between a rendered image and a reference image. The reference image can be the source image, or another rendered image. A difference image can represent a simple subtractive difference between a rendered image and a reference image. Or it can represent a magnified subtractive difference. A difference image can also represent a least squares difference between the colors of a rendered image and a reference image. Or, it can be a topographical representation of the differences between a rendered image and a reference image. The topographical representation can have a color basis.

The rendering intent selection engine receives user input selecting a previewed rendered image. It identifies the rendering intent associated with the received rendered images and selects it to render the received document or image. The rendering intent selection engine can also receive user input selecting a previewed difference of rendered images. If the reference image in the received difference is the source image, the selection engine identifies the rendering intent associated with the received difference and selects it to render the received document or image. If the reference image in the received difference is another rendered image, the selection engine prompts a user to select a rendering intent from the two rendering intents used to create the rendered images which were used to create the rendered difference image.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a schematic illustration of a rendering intent selection engine.

FIG. 2 is a flow chart depicting a method for selecting a plurality of rendering intents to be used to map the colors of an image from one color gamut to another color gamut.

FIG. 3 is a flow chart depicting a method for contrasting rendering intents by choosing to simultaneously display the rendered images, differences between the rendered images, or differences between a rendered image and the given image.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

As shown in Fig. 1, a Rendering Intent Selection Engine 50 receives a color document or image in a Receive Color Image Module 100, and a set of rendering intents in a Receive Rendering Intent Module 200. Selection Engine 50 creates a set of rendered images by rendering the received image according to the set of received rendering intents, and provides a user with a set of contrasted images in Contrast Rendered Images Module 300. Selection Engine 50 permits users to select one of the contrasted images, and thereby receives a rendering intent selection in Receive Selection Module 400.

Selection Engine 50 receives a source color image or document in Receive Image Module 100. The source color document, or image received, contains colors found entirely within the color gamut of a source color gamut. For example, the received image may be a scanned photograph containing only those colors found within the color gamut of the scanner used to scan the photograph. Or the received image may be a bitmap created and displayed on a color monitor, and may therefore contain only those colors found within the color gamut of the display.

Receive Image Module 100 may receive either an entire color image or document, or only a portion of a color image or document, and may prompt and receive commands from a user to display either an entire image or only a portion thereof. For example, Receive Image Module 100 may receive a ten page document, prompt a user to render the entire document or only a portion thereof, and receive user input to render only the second page of the document. The portion of a document or image received by Receive Image Module 100 can be selected in a number of ways well known in the art. For example, a user can select a portion by defining its vertices or edges. Similarly, a user can select a portion by highlighting it with a computer mouse according to well known methods developed in

programs having graphical user interfaces. These, and other methods of selecting an entire document, image, or portion thereof, are within the scope of the present invention.

Selection Engine 50 also receives a set of rendering intents in Receive Rendering Intent Module 200. In one implementation, Receive Intent Module 200 receives a set of all known rendering intents. These include perceptual, saturation, absolute colorimetric, and relative colorimetric. In another implementation, Receive Intent Module 200 receives a subset of all known rendering intents. In yet another implementation Receive Intent Module 200 displays a list of all known rendering intents, prompts a user to select a number of intents from the list, and receives the set of rendering intents selected from the list.

As shown in Fig. 2, in one implementation Receive Intent Module 200 receives and displays a list of all known rendering intents at step 201, together with a list of associated checkboxes. Initially all of the check boxes are selected, however Receive Intent Module 200 allows a user to deselect undesired rendering intents by deselecting their associated checkboxes. At step 202, Receive Intent Module 200 prompts the user to accept the displayed set of selected rendering intents or to preview all available rendering intents. If the user accepts the set of selected rendering intents, Receive Intent Module 200 exits at step 209.

However, if the user elects to preview all available rendering intents, Receive Intent Module 200 initializes a rendering intent list at step 203. In one implementation, the initialization step includes obtaining memory to hold a rendering intent list, initializing the memory to hold a null list, and selecting a first rendering intent as a current rendering intent. Next, Receive Intent Module 200 loops through all known rendering intents in steps 204 through 208, and compiles a list of rendering intents to be used in step 206.

In step 204, Receive Intent Module 200 renders and displays the received image according to the current rendering intent. At step 205, Receive Intent Module 200 prompts the user to accept or reject the current intent. If the user accepts the intent, Receive Intent Module 200 updates a list of received rendering intents at step 206 before proceeding to step 207. If the user rejects the intent, Receive Intent Module 200 proceeds directly to step 207. At step 207, Receive Intent Module 200 determines if the current rendering intent is the last rendering intent. If it is, Receive Intent Module 200 stops execution at step 209. If it is not, Receive Intent Module 200 selects the next known rendering intent at step 208, and reenters

the loop for compiling a list of rendering intents by rendering and displaying the received image according to the newly selected current rendering intent at step 204.

As shown in Fig. 1, Selection Engine 50 contrasts a set of rendered images in Contrast Images Module 300. In one implementation, Contrast Images Module 300 contrasts rendered images by simultaneously previewing the rendered images. In another implementation, Contrast Images Module 300 contrasts rendered images by simultaneously previewing rendered image differences.

As shown in Fig. 3, in one implementation, Contrast Images Module 300 prompts a user to enter input to contrast either rendered images or rendered differences at step 301. If the user elects to contrast rendered images at step 302, Contrast Images Module 300 contrasts the set of rendered images by previewing them, and exits at step 309. In one implementation, the set of rendered images are previewed by simultaneously displaying them on a monitor. In another implementation, the set of rendered images are previewed by simultaneously displaying their color values, e.g. their RGB pixel values, on a monitor. In yet another implementation, the set of rendered images are previewed by simultaneously printing them on a single sheet of paper. Still other implementations are possible, and within the scope of the claimed invention.

Contrast Images Module 300 may also contrast a set of rendered images by previewing a set of rendered image differences. If a user elects to contrast rendered image differences at step 302, Contrast Images Module 300 initializes a list of rendered differences at step 303. In one implementation, the initialization step includes calculating a list of possible rendered differences, obtaining memory to hold a list of rendered differences, initializing the memory to hold a null list, and selecting a current rendered difference from the list of possible rendered differences. Rendered differences are constructed between a rendered image, and a reference image. In one implementation, the reference image is the received source image. In another implementation, the reference image is another rendered image.

Contrast Images Module 300 can create rendered differences in a number of ways. In one implementation Contrast Images Module 300 creates a simple difference image by subtracting the color values of a reference image from the rendered image. In another implementation, Contrast Images Module 300 creates a least squares difference image by

computing for each pixel in the image the sum of the squares of the differences between the color values of the associated pixels in the rendered and reference images. Still other implementations are possible and within the scope of the invention. For example, the values in the difference image can be magnified by multiplying the computed differences by a magnification factor.

In many instances, the differences between the rendered image and reference image are small, and the difference image appears mostly black when displayed on a monitor or printed on a color printer. Thus, in one implementation, Contrast Images Module 300 is configured to create a difference image which is a topographical map of the differences between a rendered and reference image. Further, in one implementation the topographical map is colorized. For example, in one implementation Contrast Images Module 300 creates a a colorized topographical map based on the least squared differences between a rendered image and a reference image. In the topographical map, pixels in the difference image are assigned unique colors based on the least squares difference between the color values of the corresponding pixels in the rendered and reference images. For example, where the least squares difference between a pixel in the rendered and reference images lies in one range, e.g. 0 to 5, the corresponding pixel in the difference image is represented by an associated color, e.g. red. And where the least squares difference lies in another range, e.g. 5 to 10, the corresponding pixel in the difference image is represented by another associated color, e.g. green.

Other methods of creating topographical difference images are possible, and still within the scope of the invention. For example, difference metrics other than the least squares difference metric can be used, such as a weighted least squares difference metric, or a simple difference metric. Or a larger or smaller range of difference values can be associated with a color used in the topographical map. For example, red can be associated with a least squares difference range from 0 to 20, or from 0 to 3. Or different colors can be associated with the ranges of difference values. For example, blue can be associated with difference values ranging between 0 and 5, rather than red. These and similar variations are within the scope of the invention.

As shown in Fig. 3, following initialization step 303, Contrast Images Module 300 loops through the list of possible rendered differences in steps 304 through 308, and compiles

a list of differences to be contrasted at step 306. In step 304, Contrast Images Module 300 creates and displays a current rendered difference from the list of possible differences, and then prompts a user to accept or reject the difference at step 305. If the user accepts the difference, Contrast Images Module 300 updates the list of differences at step 306 before proceeding to step 307. If the user rejects the difference, Contrast Images Module 300 proceeds directly to step 307. At step 307, Contrast Images Module 300 determines if the current rendered difference is the last possible rendered difference. If it is, Contrast Images Module 300 exits execution at step 309. If it is not, Contrast Images Module 300 selects the next possible rendered difference as the current difference at step 308, and reenters the loop for compiling a list of selected differences by creating and displaying the current difference at step 304.

In one implementation, the set of rendered differences compiled in step 306 are previewed by simultaneously displaying them on a monitor. In another implementation, the set of rendered differences are previewed by simultaneously displaying their color values, e.g. RGB pixel values, on a monitor. In yet another implementation, the set of rendered differences are previewed by simultaneously printing them on a printer. Still other implementations are possible, and within the scope of the invention.

As shown in Fig. 1, Receive Selection Module 400 receives a contrasted rendered image from Contrast Images Module 300. In one implementation, Contrast Images Module 300 simultaneously displays rendered images or differences on a monitor. A user selects one of the rendered images or differences from the monitor, e.g. by positioning a mouse over the image and selecting it with a mouse click. In another implementation, Contrast Images Module 300 displays a list of rendered images or differences, together with an associated checklist. A user selects a desired rendered image or difference by checking the appropriate check box. In yet another implementation, Contrast Images Module 300 displays the rendered images together with an identification tag, such as a number, letter, or name, and prompts the user to enter a rendered image selection. For example, a user could enter the number 5 at a keyboard to select the rendered image labeled 5. These and other implementations are within the scope of the present invention.

Receive Selection Module 400 identifies the selected rendered image or difference, and notifies Selection Engine 50. Selection Engine 50 receives the contrasted rendered

image or difference, identifies the received rendering intent corresponding to the received rendered image or difference, and selects it to render the color image. When Selection Engine 50 receives a rendered difference, it first identifies the reference image as being the source image or another rendered image. If the reference image is the source image, Selection Engine 50 identifies the selected rendering intent as the intent used to create the rendered image. If the reference image is another rendered image, Selection Engine 50 prompts a user to identify one of the two rendering intents used to create the two rendered images as the selected rendering intent.

This invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

To provide for interaction with a user, the invention can be implemented on a computer system having a display device such as a monitor or LCD screen for displaying information to the user and a keyboard and a pointing device such as a mouse or a trackball by which the user can provide input to the computer system. The computer system can be programmed to provide a graphical user interface through which computer programs interact with users.

The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, the steps of the invention can be performed in a different order and still achieve desirable results.

What is claimed is:

07844-4 01/P387